

PB-227 775

LNEPLT

Paul R. Dorn

Federal Water Pollution Control Administration
Annapolis, Maryland

August 1969

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

PB 227 775

Chesapeake Technical Support Laboratory
Middle Atlantic Region
Federal Water Pollution Control Administration
U.S. Department of the Interior

Technical Report No. 21

LNEPLT ✓

by

Paul R. Dorn

August 1969

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

Johan A. Aalto, Chief, CTSL
Norbert A. Jaworski, Chief, Engineering Section
Richard Burkett, Draftsman

I

LNEPLT was written to aid the Chesapeake Technical Support Laboratory of the Middle Atlantic Region, Federal Water Pollution Control Administration in analyzing and displaying data.

The collection of subroutines uses a modified version of the subroutine PLOT written by the computing center of the Johns Hopkins University.

LNEPLT is operational on the IBM 360/Model 65 at the United States Geological Survey computer center, Department of the Interior in Washington, D.C. This package was compiled using the IBM G-level compiler. The subroutines should work with little or no change on any 360 IBM computer with a G or H level compiler.

For further information regarding this and other available plotting routines, contact, Dr. Norbert Jaworski, CTSL, Annapolis Science Center, Annapolis, Maryland 21401, or phone 1-301-268-5038.

II

TABLE OF CONTENTS

	Page
ABSTRACT	1
DESCRIPTION	2
SUBROUTINES	3
RESTRICTIONS	16
TIMING	17
STORAGE REQUIREMENTS	17
APPENDIX I: PROGRAM LISTING	18
APPENDIX II: SAMPLE MAIN PROGRAM AND SAMPLE PROBLEM	26

III

PROGRAM: LNEPLT

ABSTRACT: LNEPLT is a collection of subroutines which enables the user, with a minimum of programming, to create a plot of one or two dependent variables versus a common independent variable on the 1403 line printer.

DESCRIPTION:

LNEPLT consists of three subroutines (with multiple entries) which allow the user to plot on the 1403 line printer one or two dependent variables (Y-variables) versus a common X-scale. The normal size of the plot is 101 printing spaces along the X-axis and 51 lines in the Y-direction. However, the size of the Y-axis may be changed if desired.

The axes' scales are normally calculated using the maximum and minimum values of the points to be plotted. This is done by the plotting subroutine itself. When plotting two Y-variables, ID's are given to identify the different variables; this feature is not available when only one Y-variable is plotted.

A description of the subroutines in the package, and their various entries, follows. Each entry is considered a separate subroutine (and to the programmer, they act as separate routines.)

SUBROUTINE TWOPLT

Usage:

```
CALL TWOPLT (NSP1,NSP2,X1,Y1,X2,Y2,M,IHEAD)
```

Purpose:

This subroutine is used to plot two Y-variables versus a common X-scale. The plot is labeled with a one line title. Each graph begins on a new page. However, at the end of the plot, the routine does not skip to the next page. The coordinates of the points are re-arranged in descending Y-order. However the pairs [X(I),Y(I)] remain together (i.e. the X's are interchanged as well as the Y's). The two sets of lists (corresponding to Y-variables one and two) remain distinct.

List of Variables:

- NSP1 - The number of points of variable Y1 to be plotted.
- NSP2 - The number of points of variable Y2 to be plotted.
- X1 - The vector containing the X-coordinates of the first Y-variable to be plotted (Y1).
Dimension $\leq$ NSP1.
- Y1 - The vector containing the Y-coordinates of first Y-variable to be plotted.
Dimension $\leq$ NSP1.

- X2 - The vector containing the X-coordinates of the second Y-variable to be plotted (Y2).
Dimension $\leq$ NSP2.
- Y2 - The vector containing the Y-coordinates of the second Y-variable to be plotted.
Dimension $\leq$ NSP2.
- M - If zero, separate Y-scales are calculated for Y1 and Y2. If one, a common Y-scale is used.

IHEAD- The title of the graph. Dimension = 20.

NSP1, NSP2, M and IHEAD are INTEGER*4; all others are REAL*4.

SUBROUTINE PLTTWO

Usage:

CALL PLTTWO (NSP1, NSP2, X1, Y1, X2, Y2, M)

Purpose:

PLTTWO plots two Y-variables versus a common X-variable. This subroutine is identical to TWOPLT except it assumes the user has supplied a title and (if desired) skipped to a new page. To conform with the output of TWOPLT, the user has three lines for the title (TWOPLT gives a one line title and skips two lines before beginning the actual plot).

The coordinates are the points are re-arranged in descending Y-order. However the pairs [X(I),Y(I)] remain together (i.e. the X's are interchanged as well as the Y's). The two sets of lists (corresponding to Y-variables one and two) remain distinct.

List of variables:

- NSP1 - The number of points of variable Y1 to be plotted.
- NSP2 - The number of points of variable Y2 to be plotted.
- X1 - The vector containing the X-coordinates of the first Y-variable to be plotted (Y1).
Dimension $\leq$ NSP1.

- Y1 - The vector containing the Y-coordinates of the first Y-variable to be plotted.
Dimension $\leq$ NSP1.
- X2 - The vector containing the X-coordinates of the second Y-variable to be plotted (Y2).
Dimension $\leq$ NSP2.
- Y2 - The vector containing the Y-coordinates of the second Y-variable to be plotted.
Dimension $\leq$ NSP2.
- M - If zero, separate Y-scales are calculated for Y1 and Y2. If one, a common Y-scale is used.

NSP1, NSP2 and M are INTEGER*4. All others are REAL*4.

SUBROUTING ONEPLT

Usage:

```
CALL ONEPLT (NSP1, X1, Y1, IHEAD)
```

Purpose:

This subroutine is used to plot one Y-variable versus an X-variable. The plot is labeled with one line of title. Each graph begins on a new page. However, at the end of the plot, the routine does not skip to the next page.

The coordinates of the points are re-arranged in descending Y-order. However the pairs [X(I),Y(I)] remain together (i.e. the X's are reordered as well as the Y's).

List of Variables:

- NSP1 - The number of points to be plotted.
- X1 - The vector containing the X-coordinates of the points to be plotted.
Dimension $\leq$ NSP1.
- Y1 - The vector containing the Y-coordinates of the points to be plotted.
Dimension $\leq$ NSP1.
- IHEAD - The title of the graph. Dimension = 20.

NSP1 and IHEAD are INTEGER*4; X1 and Y1 are REAL*4.

SUBROUTINE PLTONE

Usage:

CALL PLTONE (NSP1, X1, Y1)

Purpose:

This subroutine is used to plot one Y-variable versus an X-variable. The plot is the same as from ONEPLT except the user must supply the title and the paging (if desired).

The coordinates of the points are re-arranged in descending Y-order. However the pairs [X(I),Y(I)] remain together (i.e. the X's are interchanged as well as the Y's).

List of Variables:

- NSP1 - The number of points to be plotted.
- X1 - The vector containing the X-coordinates of the points to be plotted.
- Y1 - The vector containing the Y-coordinates of the points to be plotted.

NSP1 is INTEGER*4; X1 and Y1 are REAL*4.

The previous four subroutines are the ones the user employs to do the plotting. The basic subroutine of the package is PLOT1. The subroutine is not directly accessed by the programmer. It is used by the previous routines to plot a symbol in a specific location.

Initialization of the program is done using the pseudo-subroutine BLOCK DATA. This subroutine is not directly referenced by the program.

The following subroutines are employed to change certain characteristics of the plotter. Once these changes have been made, the new parameters are used until further use is made of these utility routines. That is, once a plot has been made, the program does not reset its parameters back to their initial values.

SUBROUTINE LNECHG

Usage:

CALL LNECHG (LINE)

Purpose:

This subroutine changes the number of lines in the Y-axis. This is initially set at 50 (actually 51 since a zero line is plotted). Calling LNECHG (LINE) sets the Y-axis as LINE+1 lines. Calling LNECHG (50) sets the Y-axis back to normal. At this position, the program completely fills one computer page. LINE should be a multiple of 10. If more than 50 lines are specified, the user must insure that the page overflow test is suppressed.

List of Variables:

LINE - The number of lines in the Y-axis
(Actually the number of lines minus one).

LINE is INTEGER*4.

SUBROUTINE SYMCHG

Usage:

CALL SYMCHG (EXTRA, NUMBER)

Purpose:

This subroutine replaces the symbols used in plotting. Originally the plotting symbol for Y1 is a dot (.). The plotting symbol for Y2 is normally an asterisk (*). If both symbols should appear in the same printing location, an oh (O) is plotted.

These symbols should not be used as replacements:

+ 1 -

since they are used to draw the graph's axes.

List of Variables:

EXTRA - This is the replacement symbol.

NUMBER - This variable specifies which symbol is to be replaced, coded as follows:

- 1 - Replaces the plotting symbol for Y1.
- 2 - Replaces the plotting symbol for Y2.
- 3 - Replaces the plotting symbol used to show that variables Y1 and Y2 both occupy the same printing location.

NUMBER is INTEGER*4 while EXTRA is INTEGER*2.

SUBROUTINE NAMCHG

Usage:

```
CALL NAMCHG (NAME, I)
```

Purpose:

This subroutine changes the names of the Y-variables.
They are used only when plotting two Y-variables.
Originally the two names are 'VAR 1 ' and
'VAR 2 '.

List of Variables:

NAME - This variable gives the replacement name.
Its printing format is 2A4. Dimension = 2.

I - This is the number of the Y-variable whose
name is to be changed (either 1 or 2).

Both NAME and I are INTEGER*4.

SUBROUTINE GIVMM

Usage:

CALL GIVMM (I)

Purpose:

This subroutine is used to signify the user is supplying the axes' scales. If this routine is never called, it is the same as calling GIVMM(0).

List of Variables:

I - If zero, the program determines the axes' scales.
If one, the user must supply the axes' scales.
(See subroutine MAXMIN.)

I is of type INTEGER*4.

SUBROUTINE MAXMIN

Usage:

```
CALL MAXMIN (I1, I2, I3, I4, I5, I6, XMIN, XMAX,  
            Y1MIN, Y1MAX, Y2MIN, Y2MAX)
```

Purpose:

This subroutine gives the user-selected maxima and minima of the X and Y scales. To use the supplied values of maxima and minima requires a call to GIVMM(1). (See SUBROUTINE GIVMM.)

When using this option, note that the program either calculates all its own scales or uses all the given ranges--a combination of the two is not permitted. However, if the coordinate of a point to be plotted is outside the selected range, the normal method of selection is used to calculate the offending range value. Therefore by selecting obviously incorrect ranges, certain scales may be calibrated by the program and others by the user (e.g. setting XMIN = +9.E+40 assures that the subroutine will recalculate XMIN).

List of Variables:

- I1 - If zero, a new value for XMIN is not given.
If one, a new value of XMIN is given.
- I2 - If zero, a new value for XMAX is not given.
If one, a new value for XMAX is given.
- I3 - If zero, a new value for Y1MIN is not given.
If one, a new value for Y1MIN is given.
- I4 - If zero, a new value for Y1MAX is not given.
If one, a new value for Y1MAX is given.
- I5 - If zero, a new value for Y2MIN is not given.
If one, a new value for Y2MIN is given.
- I6 - If zero, a new value for Y2MAX is not given.
If one, a new value for Y2MAX is given.
- YMIN - The new value for XMIN (requires I1 = 1).
- XMAX - The new value for SMAX (requires I2 = 1).
- Y1MIN - The new value for Y1MIN (requires I3 = 1).
- Y1MAX - The new value for Y1MAX (requires I4 = 1).
- Y2MIN - The new value for Y2MIN (requires I5 = 1).
- Y2MAX - The new value for Y2MAX (requires I6 = 1).

I1 - I6 are INTEGER*4; the others are of type REAL*4.

Initially all the range values are set to 0.0. See

SUBROUTINE GIVMM(I) for further instructions.

RESTRICTIONS:

The following names may not be referenced by the programmer except as previously described:

GIVMM

LNECHG

LNEPLT

MAXMIN

NAMCHG

ONEPLT

PLOT1

PLTONE

PLTTWO

SYMCHG

TWOPLT

TIMING:

To plot one graph consisting of two Y variables and 25 points for each variable took 0.60 seconds.

(See the sample problem in Appendix II.)

STORAGE REQUIREMENTS:

This subroutine package requires 686_{16} bytes.

APPENDIX I: PROGRAM LISTING

LINEPT

```

0
PLT 10
PLT 30
PLT 30
PLT 40
PLT 50
PLT 60
PLT 70
PLT 80
PLT 90
PLT 100
PLT 110
PLT 120
PLT 130
PLT 140
PLT 150
PLT 160
PLT 170
PLT 180
PLT 190
PLT 200
PLT 210
PLT 220
PLT 230
PLT 240
PLT 250
PLT 260
PLT 270
PLT 280
PLT 290
PLT 300
PLT 310
PLT 320
PLT 330
PLT 340
PLT 350
PLT 360
PLT 370
PLT 380
PLT 390
PLT 400

PLOTING SUBROUTINE WHICH PLOTS DATA OF TWO DEPENDENT VARIABLES
USING THE SAME ARCSISSA.

SUBROUTINE TWOPLT(NSP1,NSP2,X1,Y1,X2,Y2,M,HEAD)
REAL X1(1),X2(1),Y1(1),Y2(1),XTEMP(11)
INTEGER HEAD(20),NAME1(2),NAME2(2),NAME1(2)
INTEGER*2 A(101),B(101)/101*1H-/,PLUS/1H+/,BLANK/1H /,SYMBOL(3),ON
E/1H1/,EXTRA
COMMON/LNEPLT/NAME1,NAME2,L,LINE,XMIN1,XMAX1,YMIN2,YMAX2,Y2MIN2,Y2
MAX2,IOPT1,SYMBOL
WRITE(6,1000) HEAD
FORMAT(1H1,9X,20A4)
ENTRY PLTTWO(NSP1,NSP2,X1,Y1,X2,Y2,M)
NSP2M1 = NSP2 - 1
KPLT = 2
GO TO 200
ENTRY ONEPLT(NSP1,X1,Y1,HEAD)
WRITE(6,1000) HEAD
ENTRY PLTONE(NSP1,X1,Y1)
M = 0
Y2MAX = -9.E20
Y2MIN = 9.E20
KPLT = 1
NSP2 = 0
WRITE(6,1010)
FORMAT(1H0)
NSP1M1 = NSP1 - 1
XMAX = X1(NSP1)
XMIN = X1(NSP1)
DO 1 I=1,NSP1M1
  IP1 = I+1
DO 2 J=IP1,NSP1
  IF (Y1(J).LE.Y1(I)) GO TO 2
  SORT THE Y1 ARRAY INTO DESCENDING ORDER
  T = Y1(I)
  Y1(I) = Y1(J)
  Y1(J) = T

```

```

2      T = X1(I)
      X1(I) = X1(J)
      X1(J) = T
      CONTINUE
1      XMIN = AMIN1(XMIN,X1(I))
      XMAX = AMAX1(XMAX,X1(I))
      IF (KPLT.EQ.1) GO TO 210
      XMIN = AMIN1(XMIN,X2(NSP2))
      XMAX = AMAX1(XMAX,X2(NSP2))
      DO 3 I=1,NSP2M1
        IPI = I+1
      DO 4 J=IPI,NSP2
        IF (Y2(J).LE.Y2(I)) GO TO 4
        T = Y2(I)
        Y2(I) = Y2(J)
        Y2(J) = T
      T = X2(I)
        X2(I) = X2(J)
        X2(J) = T
      CONTINUE
4      XMIN = AMIN1(XMIN,X2(I))
      XMAX = AMAX1(XMAX,X2(I))
      XMIN = AMIN1(X1(NSP1), X2(NSP2), XMAX)
      XMIN = AMIN1(X1(NSP1), X2(NSP2), XMIN)
      C
      C
      C
      C
      AT THIS POINT, THE Y ARRAYS ARE ORDERED IN DESCENDING ORDER
      AND THE ABSCISSA IS BOUNDED BY (XMIN,XMAX)
      V2MAX = V2(I)
      V2MIN = V2(NSP2)
      V1MAX = V1(I)
      V1MIN = V1(NSP1)
      IF (IOPT1.EQ.0) GO TO 30
      XMIN = AMIN1(XMIN,XMIN1)
      XMAX = AMAX1(XMAX,XMAX1)
      V1MIN = AMIN1(V1MIN,V1MIN2)
      V1MAX = AMAX1(V1MAX,V1MAX2)
      IF (KPLT.EQ.1) GO TO 30
      V2MIN = AMIN1(V2MIN,V2MIN2)
      V2MAX = AMAX1(V2MAX,V2MAX2)
      CONTINUE
30      IF (MONE.1) GO TO 5
      T = AMIN1(V1MIN,V2MIN)
      V1MIN = T
      V2MIN = T
      T = AMAX1(V1MAX,V2MAX)

```

PLT 400
 PLT 410
 PLT 420
 PLT 430
 PLT 440
 PLT 450
 PLT 460
 PLT 470
 PLT 480
 PLT 490
 PLT 500
 PLT 510
 PLT 520
 PLT 530
 PLT 540
 PLT 550
 PLT 560
 PLT 570
 PLT 580
 PLT 590
 PLT 600
 PLT 610
 PLT 620
 PLT 630
 PLT 640
 PLT 650
 PLT 660
 PLT 670
 PLT 680
 PLT 690
 PLT 700
 PLT 710
 PLT 720
 PLT 730
 PLT 740
 PLT 750
 PLT 760
 PLT 770
 PLT 780
 PLT 790
 PLT 800
 PLT 810
 PLT 820
 PLT 830
 PLT 840
 PLT 850

```

Y2MAX = Y
Y1MAX = Y
5 IF (KPLT, EQ, 2) WRITE(6, 1020) (SYMBOL(1), I=1, 5), NAME1, NAME2, (SYMBOL(
1020 2), I=1, 5)
FORMAT(1H0, 3X, 5A1, 3H = , 2A4, T112, 2A4, 3H = , 5A1)
XSCAL = (XMAX - XMIN) / 100.
Y1SCAL = (Y1MAX - Y1MIN) / FLOAT(LINE)
Y2SCAL = (Y2MAX - Y2MIN) / FLOAT(LINE)
WRITE (6, 8) (I, I=10, 100, 10)
FORMAT (1HC, 14X, 1HC, 10(7X, 13))
DO 220 I=1, 101, 10
B(I) = PLUS
DO 9 I=1, 101
A(I) = B(I)
I = Y1 SIGNAL NUMBER
J = Y2 SIGNAL NUMBER
K = NUMBER OF LINES PRINTED
I = 1
J = 1
K = 0
10 CALL PLOT1(A, ONE, ONE, ONE, 1)
CALL PLOT1(A, ONE, ONE, ONE, 101)
Y = .5 * FLOAT(K)
IF (I, GT, NSP1) GO TO 14
ELOWER = Y1MAX - Y1SCAL
EUPPER = ELOWER + Y1SCAL
IF (Y1, GT, EUPPER, OR, Y1, LT, ELOWER) GO TO 14
N = (N1 - XMIN) / XSCAL + 1.5
CALL PLOT1(A, SYMBOL(1), SYMBOL(2), SYMBOL(3), N)
I = I + 1
GO TO 13
IF (J, GT, NSP2) GO TO 15
ELOWER = Y2MAX - Y2SCAL
EUPPER = ELOWER + Y2SCAL
IF (Y2(J), GT, EUPPER, OR, Y2(J), LT, ELOWER) GO TO 15
N = (X2(J) - XMIN) / XSCAL + 1.5
CALL PLOT1(A, SYMBOL(2), SYMBOL(1), SYMBOL(3), N)
J = J + 1
GO TO 14
READY NOW TO PLOT
C
C
C
15 L = LINE - K
IF (MOD(L, 10), EQ, 0) GO TO 17
WRITE (6, 18) A

```

PLT 860
 PLT 870
 PLT 880
 PLT 890
 PLT 900
 PLT 910
 PLT 920
 PLT 930
 PLT 940
 PLT 950
 PLT 960
 PLT 970
 PLT 980
 PLT 990
 PLT 1000
 PLT 1010
 PLT 1020
 PLT 1030
 PLT 1040
 PLT 1050
 PLT 1060
 PLT 1070
 PLT 1080
 PLT 1090
 PLT 1100
 PLT 1110
 PLT 1120
 PLT 1130
 PLT 1140
 PLT 1150
 PLT 1160
 PLT 1170
 PLT 1180
 PLT 1190
 PLT 1200
 PLT 1210
 PLT 1220
 PLT 1230
 PLT 1240
 PLT 1250
 PLT 1260
 PLT 1270
 PLT 1280
 PLT 1290
 PLT 1300
 PLT 1310

```

18  FORMAT(15X,101A1)
    GO TO 19
17  YTEMP = YMIN+VISCAL*FLOAT(L)
    YTEMP2 = Y2*IN + Y2SCAL*FLOAT(L)
    WRITE(6,29) YTEMP, A, YTEMP2
29  FORMAT(1PE13.3,2X,101A1,1PE11.3)
    IF(L.EQ.0) GO TO 26
    IF (L.EQ.1) GO TO 21
    DO 22 I=1,101
      A(I) = BLANK
      K = K+1
    GO TO 10
21  DO 24 I=1,101
    24  A(I)=B(I)
      K = K+1
    GO TO 10
26  XTEMP(I) = XMIN
    DO 102 I=2,11
    102 XTEMP(I)=XTEMP(I-1)*XSCAL*10.
    WRITE(6,101) (XTEMP(I),I=1,11,2), (XTEMP(I),I=2,10,2)
    101  FORMAT(10X,6(1PE10.3,10X)/20X,5(1PE10.3,10X))
    RETURN
    ENTRY LNECHG(IJ)
    LINE=IJ
    GO TO 70
    ENTRY SYNCHG(EXTRA,IJ)
    SYMBOL(IJ) = EXTRA
    99  10 10
    ENTRY NAMEHG(NAMET,IJ)
    DO 60 I=1,2
    60  GO TO (40,50),IJ
    40  NAME(I) = NAMET(I)
    GO TO 60
    50  NAME2(I)=NAMET(I)
    60  CONTINUE
    GO TO 70
    ENTRY MAXMIN(I1,I2,I3,I4,I5,I6,X11,X22,Y11,Y12,Y21,Y22)
    IF(I1.NE.0) XMIN1 = X11
    IF(I2.NE.0) XMAX1 = X22
    IF(I3.NE.0) YMIN2 = Y11
    IF(I4.NE.0) YMAX2 = Y12
    IF(I5.NE.0) YMIN2 = Y21
    IF(I6.NE.0) YMAX2 = Y22
    GO TO 70
    ENTRY GIVM(I)
    IOPTI=I

```

PLT 1320
 PLT 1330
 PLT 1340
 PLT 1350
 PLT 1360
 PLT 1370
 PLT 1380
 PLT 1390
 PLT 1400
 PLT 1410
 PLT 1420
 PLT 1430
 PLT 1440
 PLT 1450
 PLT 1460
 PLT 1470
 PLT 1480
 PLT 1490
 PLT 1500
 PLT 1510
 PLT 1520
 PLT 1530
 PLT 1540
 PLT 1550
 PLT 1560
 PLT 1570
 PLT 1580
 PLT 1590
 PLT 1600
 PLT 1610
 PLT 1620
 PLT 1630
 PLT 1640
 PLT 1650
 PLT 1660
 PLT 1670
 PLT 1680
 PLT 1690
 PLT 1700
 PLT 1710
 PLT 1720
 PLT 1730
 PLT 1740
 PLT 1750
 PLT 1760
 PLT 1770

PAGE 6

PLT 178C
PLT 179070 RETURN
END

***** END OF PROGRAM *****

PLOT1

```

SUBROUTINE PLOT1(A,SYM1,SYM2,SYM3,P(ACE)
INTEGER*2 SYM1,SYM2,SYM3,A
INTEGER PLACE
DIMENSION A(1)
IF(PLACE.LE.0) GO TO 10
IF(A(PLACE).EQ.SYM3) RETURN
IF(A(PLACE).EQ.SYM2) GO TO 30
A(PLACE) = SYM1
RETURN
10 WRITE(6,20) PLACE
20 FORMAT(1H1,'TRIED TO PLACE A SYMBOL IN SPACE NUMBER ',I3)
STOP
30 A(PLACE) = SYM3
RETURN
END

```

PT1 0
PT1 10
PT1 20
PT1 30
PT1 40
PT1 50
PT1 60
PT1 70
PT1 80
PT1 90
PT1 100
PT1 110
PT1 120
PT1 130
PT1 140

***** END OF PROGRAM *****

25

PAGE 8

BLOCK DATA

```
BLOCK DATA
COMMON/LNEPLT/NAME1,NAME2,LINE,XMINMX(6),IOPT1,SYMBOL
INTEGER*2 SYMBOL(3)/IN,IM,IMO/
REAL XMINPX/6*0.0/
INTEGER NAME1(2)/4HVAR *4H1 /,NAME2(2)/4HVAR *4H2 /.LINE/60/,I
$ OPT1/3/
END
```

BLK 0
BLK 10
BLK 20
BLK 30
BLK 40
BLK 50
BLK 60

***** END OF PROGRAM *****

APPENDIX II: SAMPLE MAIN PROGRAM AND SAMPLE PROBLEM

XYPLOT - MAIN ROUTINE (SAMPLE)

```

C C C SAMPLE MAIN PROGRAM FOR LNEPLT
C C C OPTIONS -
C C C 1 - IF ZERO, EXPECT TWO Y-VARIABLES
C C C     IF ONE, ONLY ONE Y-VARIABLE ,
C C C 2 - IF ZERO, NO ACTION TAKEN
C C C     IF ONE, RANGE VALUES READ IN
C C C 3 - IF ZERO, 50 LINES/PLT IN Y DIRECTION
C C C     IF ONE, READ IN LINE COUNT
C C C 4 - IF ZERO, SEPERATE Y-SCALES CALCULATED
C C C     IF ONE, COMMON Y-SCALE USED
C C C 5 - 10 NOT USED
C C C .....
C C C DIMENSION X1(100),Y1(100),X2(100),Y2(100),IOPT(10),IYIT(20)
C C C $ READ(5,500,END=120) IYIT,IOPT,XMIN,XMAX,YIMIN,YIMAX,
C C C     Y2MIN,Y2MAX,MCOUNT,N
C C C 900 FORMAT(20A4/10I1,6G10.6,15,15)
C C C RESET INITIAL VALUES
C C C CALL GIVMM(0)
C C C CALL LNECHG(50)
C C C RESET OPTIONS AS DESIRED
C C C RANGE VALUES
C C C IF(IOPT(2)) 30,30,20
C C C 20 CALL GIVMM(1)
C C C     CALL MAXMIN(1,1,1,1,1,1,XMIN,XMAX,YIMIN,Y1,MAX,Y2MIN,Y2MAX)
C C C LINE COUNTER
C C C 30 IF(IOPT(3)) 50,50,40
C C C PLT 0
C C C PLT 10
C C C PLT 20
C C C PLT 30
C C C PLT 40
C C C PLT 50
C C C PLT 60
C C C PLT 70
C C C PLT 80
C C C PLT 90
C C C PLT 100
C C C PLT 110
C C C PLT 120
C C C PLT 130
C C C PLT 140
C C C PLT 150
C C C PLT 160
C C C PLT 170
C C C PLT 180
C C C PLT 190
C C C PLT 200
C C C PLT 210
C C C PLT 220
C C C PLT 230
C C C PLT 240
C C C PLT 250
C C C PLT 260
C C C PLT 270
C C C PLT 280
C C C PLT 290
C C C PLT 300
C C C PLT 310
C C C PLT 320
C C C PLT 330
C C C PLT 340
C C C PLT 350
C C C PLT 360
C C C PLT 370
C C C PLT 380
C C C PLT 390

```

```

40 CALL LNECHG(MCOUNT)
C
C NOW READY TO READ IN DATA
C
50 DO 80 I=1,N
  IF(IOPT(1)) 60,60,70
  READ(5,910) X1(I),Y1(I),X2(I),Y2(I)
  GO TO 80
70 READ(5,910) X1(I),Y1(I)
80 CONTINUE
910 FORMAT(4G10.6)
C
C MAKE THE PLOT
C
  IF(IOPT(1)) 90,90,100
  CALL TWOPLT(N,X1,Y1,X2,Y2,IOPT(4),ITIT)
  GO TO 110
100 CALL ONEPLT(N,X1,Y1,ITIT)
C
C RETURN FOR NEXT PLOT
C
110 GO TO 10
C
C END OF DATA
C
120 STOP
END

```

***** END OF PROGRAM *****

PLT 400
PLT 410
PLT 420
PLT 430
PLT 440
PLT 450
PLT 460
PLT 470
PLT 480
PLT 490
PLT 500
PLT 510
PLT 520
PLT 530
PLT 540
PLT 550
PLT 560
PLT 570
PLT 580
PLT 590
PLT 600
PLT 610
PLT 620
PLT 630
PLT 640
PLT 650
PLT 660

28



29

TEST OF THE PLOTTING PACKAGE FOR THE 1403 LINE PLOTTER--LINEPLT

..... = VAR 1

